

# Modern Engineering

January 30 | Day 15

## Prudential Financial



# LESSON ROADMAP



AWS Cloud  
Formation

Load Balancing

AWS  
DynamoDB

AWS  
Deployment

Prudential AWS

# MEF MODULE 5 DAY 15: AWS Cloud Formation

Schedule	
9:00–10:00 am	Welcome and Warm-Up / Lab Time
10:00 am–12:00 pm	<b>GUEST: Cloud Formation Template</b>
12:00 am–12:30 pm	<b>GUEST: How Prudential Uses AWS Resources</b>
12:30–1:30 pm	Lunch
1:30–2:00 pm	<b>AWS Cloud Formation Provisioning</b>
2:00–4:50 pm	<b>AWS DynamoDB</b>
4:50–5:00 pm	Bring It Home

# LEARNING OBJECTIVES

1

Understand AWS Cloud Formation architecture

2

Articulate the benefits of logging in a cloud-based application

3

Use AWS DynamoDB to replace a Postgres SQL database in an Express application

4

Explore Prudential's Cloud Formation templates and use of AWS services





# **GUEST SPEAKER:**

## **Prudential's Cloud Formation Template**



# **GUEST SPEAKER:**

## **How Prudential Uses AWS Resources**



# Break Time

Automation and CloudFormation

---

# Intro to **AWS** CloudFormation





# Intro to CloudFormation

AWS CloudFormation is an IaC tool for managing resources on AWS.

It integrates well with most AWS services and provides a set of pre-defined resource types that can be configured.

CloudFormation works by allowing you to define **templates**. These templates can be written in YAML or JSON and are declarative: they describe the resources you want to manage and use.



# Example CloudFormation Template

**Resources:**

**MyS3Bucket:**

**Type:** 'AWS::S3::Bucket'

**Properties:**

**BucketName:** my-bucket-name

Above is a simple example template for creating an **S3 bucket in YAML**.

# Common Use Cases

We'll be using CloudFormation to handle some common use cases:

- Provisioning resources, such as VMs
- Managing security groups
- SSH Keys
- EIP

Modern Engineering

---

# Deploying to *AWS*



# The limitations imposed by ephemeral environments

- No server to log into and inspect
- All configuration must be scripted
- All apps must be able to start in an automated way
- Nominally more complex networking, endpoints, and AWS infrastructure



# The limitations imposed by asynchronous systems and microservices

- The requests and logs may not match up in a linear fashion
- Higher than expected latency
- A multitude of deployments
- Nominally more complex networking, endpoints, and AWS infrastructure

# Security Groups

## Firewalls on Steroids

- Ports and traffic can be allowed by:
  - IP
  - CIDR
  - Other security groups - this is the killer feature

# IAM Roles

## RBAC and ABAC

- IAM roles cover human users, services, and applications.
- Every service, instance, and operation has an IAM role.



Modern Engineering

---

# Criticality of GOOD Logging

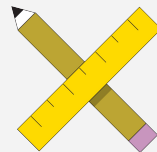






## Using CloudFormation:

- Provision an EC2 instance
- Add a security group
- Add an SSH key
- Add an EIP



**We will be available to provide additional support and troubleshoot problems.**

Detailed instructions can be found in [GitHub](#).



**Good logs in a word: **DETAIL**. You need to know not only what service failed but **have enough information to trace the path** through other services.**

# CloudWatch

- Ephemeral applications **require** centralized, external logging.
- CloudWatch logs have configurable retention.
- They are searchable.
- They span instances.

Log groups (8)	
By default, we only load up to 10000 log groups.	
<input type="text" value="Filter log groups or try prefix search"/>	
<input type="checkbox"/>	Log group
<input type="checkbox"/>	/aws/elasticbeanstalk/Vehicle-env-lb/var/log/eb-engine.log
<input type="checkbox"/>	/aws/elasticbeanstalk/Vehicle-env-lb/var/log/eb-hooks.log
<input type="checkbox"/>	/aws/elasticbeanstalk/Vehicle-env-lb/var/log/httpd/access_log
<input type="checkbox"/>	/aws/elasticbeanstalk/Vehicle-env-lb/var/log/httpd/error_log
<input type="checkbox"/>	/aws/elasticbeanstalk/Vehicle-env-lb/var/log/nginx/access.log
<input type="checkbox"/>	/aws/elasticbeanstalk/Vehicle-env-lb/var/log/nginx/error.log
<input type="checkbox"/>	/aws/elasticbeanstalk/Vehicle-env-lb/var/log/web.stdout.log
<input type="checkbox"/>	dmv

AWS Cloud Formation



# AWS DynamoDB



# AWS DynamoDB: A More Flexible Database

SQL's greatest strength can sometimes be its greatest weakness as well: the strictly controlled **structure** requires tight management of schemas and changes over time.

DynamoDB is the AWS solution to providing a NoSQL database solution: less structure, less tight relationships among schemas, just a fast and simple form of data storage.



## Guided Walk-Through: DynamoDB Demo

Exploring a Cloud-Based NoSQL Database: A Walkthrough Using [DynamoDB](#).



Solo Exercise:

## Daily Exit Ticket

3 minutes



- Please take a moment to give us your feedback after today's course!
- Use the QR code or follow the link: <https://bit.ly/pruMEFc2>



**Scan Me!**



